

برنامه‌سازی پیشرفته

تمرین کامپیووتری شماره ۵

مدرس: رامتین خسروی

طراحان: کوروش علی‌نقی، سروش صحرائی^۱

محمد عرفان دانایی، امیرعلی دهقانی، بهاره عین‌اللهی

مهلت تحویل: سه‌شنبه ۲۲ اردیبهشت ۱۴۰۴، ساعت ۵:۳۲



مقدمه

هدف از این تمرین آشنایی شما با برنامه‌نویسی شی‌عگرای رویدادمحور^۱ و استفاده از آن در کتابخانه‌های گرافیکی است. انتظار می‌رود از فنون برنامه‌نویسی که تاکنون در کلاس درس فراگرفته‌اید یا در هنگام تحویل تمرین‌ها به شما تذکر داده شده است، به طور کامل در این تمرین استفاده کنید. توصیه می‌شود تمرین را در بخش‌های کوچک جلو ببرید و هر مرحله عملکرد آن را آزمایش کنید و بررسی درستی آن را برای مرحله آخر نگذارید. برای پاسخ به این تمرین باید از کتابخانه‌ی گرافیکی SFML^۲ استفاده کنید و همچنین باید با مفاهیمی چون ساخت پروژه به صورت چندفایلی و استفاده از گیت آشنا باشید. قبل از شروع به انجام این تمرین توصیه می‌شود حتماً مستندات موجود در این [لینک](#) را مطالعه کنید.

^۱ Event Driven Programming

^۲ Simple and Fast Multimedia Library

Bloons Tower Defense



شما می‌توانید این بازی را از طریق [این لینک](#)، به صورت آنلاین بازی کنید و با محیط بازی آشنا شوید. توجه کنید که جزئیات این تکلیف با جزئیات بازی تفاوت دارد، بنابراین صورت تکلیف را با دقت مطالعه کنید.

معرفی

بازی Bloons Tower Defense، یک بازی تک نفره در سبک Tower Defense است. نقشه بازی شامل مسیریست که دشمنان شما (که بادکنک هستند!) در آن حرکت می‌کنند. هدف بازی این است که بازیکن بتواند با قرار دادن برج‌های دفاعی خود، از حرکت دشمنان جلوگیری کند. در ادامه به جزئیات بیشتر روند بازی می‌پردازیم.

نقشه بازی

نقشه بازی به صورت مستطیلی شبکه‌بندی شده³ پیاده‌سازی می‌شود که بعضی خانه‌های آن، به عنوان مسیر شناخته می‌شوند و تمامی خانه‌های دیگر، خانه‌هایی هستند که برج‌های دفاعی می‌توانند در آن‌ها قرار گیرند. بادکنک‌ها از نقطه‌ای در نقشه وارد بازی می‌شوند و از نقطه‌ای دیگر از آن خارج می‌شوند.

فایل نقشه

بازی شما باید نقشه بازی را از یک فایل در زمان اجرا بخواند. برای مسیر این فایل از آرگومان خط فرمان استفاده نکنید و یک مسیر دلخواه در نظر بگیرید. تضمین می‌شود هر نقشه، دقیقاً یک خانه شروع و یک خانه پایان دارد. این فایل شامل کاراکترهایی خاص است که هر کاراکتر، یکی از انواع خانه‌ها را در نقشه مشخص می‌کند:

نوع خانه	کاراکتر
خانه شروع مسیر	S
خانه پایان مسیر	F
مسیر	O
خانه‌هایی که امکان ایجاد برج در آن‌ها وجود دارد	-

نمونه نقشه
----- ---0000000- ---0----0- S000000--0- ---0--0--0- ---0--0--0- ---0000--0- -----F-

که مسیر آن، نمایانگر مسیر شکل زیر است:

³ Grid



نکته مهم درباره حرکت بادکنک‌ها در مسیر، این است که در صورتی که در مسیر، بادکنک به یک چهارراه رسید، همواره مسیر مستقیم را پیش می‌گیرد. تضمین می‌شود نقشه طوری داده می‌شود که با اجرای این قانون در تمام چهارراه‌ها، بتوان از نقطه شروع به نقطه پایان رسید.

روند بازی

آماده، هدف، آتش!

همانطور که گفته شد، بادکنک‌ها از نقطه شروع مسیر وارد نقشه می‌شوند و در مسیر مشخص شده، به سمت نقطه پایانی حرکت می‌کنند. در این میان، این برج‌های شما هستند که باید با شلیک گلوله به بادکنک‌ها، از رسیدن آن‌ها به نقطه پایانی جلوگیری کنند. به ازای هر بادکنکی که توسط برج‌های شما نابود می‌شود، میزان مشخصی (بسته به نوع بادکنک نابودشده) به امتیاز بازیکن اضافه می‌شود. بازیکن می‌تواند از این امتیازها برای کاشتن برج‌ها استفاده کند.

پس از رد شدن از نقطه پایانی، بادکنک از صفحه بازی محو می‌شود. به ازای هر بادکنکی که از نقطه پایانی عبور کند، میزان مشخصی از جان بازیکن کم می‌شود. بادکنک‌ها در قالب موج‌ها حمله می‌کنند.

موج حمله

دشمنان در قالب **موج‌های حمله** وارد نقشه می‌شوند. هر موج، شامل تعداد مشخصی بادکنک است که از نقطه شروع مسیر، وارد نقشه می‌شوند.

هر موج، پس از گذشت زمان مشخصی از پایان موج قبلی ظاهر می‌شود. موج اول نیز پس از گذشت همین زمان از شروع بازی روانه می‌شود. زمان پایان هر موج زمانی است که آخرین بادکنک آن موج از بین برود. در هر موج، تعداد مشخصی از انواع بادکنک‌های موجود به شما حمله می‌کنند. همچنین در هر موج، بادکنک‌ها با فاصله زمانی مشخص و یکسانی از یکدیگر وارد نقشه می‌شود.

جدا از فاصله زمانی بین هر دو موج که یک پارامتر کلی برای بازی است، هر موج نیز پارامترهای مختص به خود را دارد:

- بازه فاصله زمانی بین ورود دو بادکنک متوالی: فاصله‌ی زمانی بین ورود هر دو بادکنک متوالی در یک موج، عددی تصادفی در بازه‌ای است که در تنظیمات موج‌ها مشخص شده است.
- تعداد هر بادکنک به تفکیک انواع بادکنک (به طور مثال، ۳ بادکنک معمولی و ۱ بادکنک باردار) دقیق کنید که در یک موج، ترتیب ظاهر شدن بادکنک‌ها در نقشه باید به صورت تصادفی باشد. بازی شما، شامل تعدادی موج است که به ترتیب در فواصل زمانی ذکرشده رخ می‌دهند.

کاشتن برج‌ها

بازیکن در طول بازی می‌تواند با توجه به موجودی امتیازش، در خانه‌هایی که مخصوص ساخت برج هستند، برج بکارد. دقیق کنید که در هر خانه، حداقل یک برج قرار می‌گیرد. در حین بازی، باید در پنلی همه انواع برج‌های بازی نمایش داده شوند و همچنین برج‌هایی که کاربر امکان ساخت آنها را دارد، باید با سایر برج‌ها تفاوت بصیری داشته باشند (به طور مثال، می‌توانید از تصویر سیاه و سفید برای برج‌هایی که کاربر امتیاز کافی برای ساختشان را ندارد استفاده کنید).

بازیکن با کلیک بر روی برج موردنظر در پنل، آن برج را انتخاب کرده و سپس با کلیک بر روی یکی از خانه‌های خالی‌ای که امکان ساخت برج در آنها وجود دارد، برج را در آن خانه می‌کارد.

بعد از کاشته شدن هر برج، باید شعاع عملکرد آن به شکل یک دایره شفاف نمایش داده شود (به تصویر نقشه نمونه توجه کنید).

پایان بازی

در هر لحظه از بازی، اگر جان بازیکن صفر یا منفی شود، بازیکن بازی را می‌بازد. اما اگر بازیکن تمام موج‌ها را بدون تمام شدن جانش بگذراند، برنده بازی می‌شود. پس از پایان بازی، صفحه‌ای باید به بازیکن نمایش داده شود تا نتیجه بازی (برد/باخت) را به او اعلام کند.

موجودیت‌های بازی

توجه کنید که برای نمایش انواع مختلف برج‌ها و بادکنک‌ها، باید از نمایه‌های مختلف استفاده کنید تا برای بازیکن از یکدیگر قابل تشخیص باشند.

برج‌ها

برج‌ها ابزار دفاعی شما برای مقابله با بادکنک‌ها هستند. هر برج، هزینه ساختی دارد که از موجودی امتیاز بازیکن کم می‌شود. همچنین، دامنه آسیب و فاصله زمانی بین هر دو شلیک متوالی هر نوع برج نیز مقدار مشخصی است. توجه کنید که در تمام صورت پروژه، منظور از فاصله، فاصله اقلیدسی است.

از آنجایی که بادکنک‌ها در حال حرکت هستند، نیاز است تا سرعت حرکت گلوله‌ها را به اندازه‌ای در نظر بگیرید که پس از شلیک آن توسط برج، با احتمال بالایی به بادکنک هدف برخورد کند (نیازی به محاسبه نقطه برخورد جدید نیست).

برج‌ها انواع مختلفی دارند که هر کدام به نحوی خاص با دشمن مبارزه می‌کنند! در ادامه به معرفی این انواع می‌پردازیم.

برج معمولی



این برج، در فواصل زمانی مشخص، به سمت نزدیک‌ترین بادکنکی که در دامنه شلیک آن قرار دارد شلیک می‌کند (در صورتی که هیچ بادکنکی در دامنه شلیک برج نباشد، شلیک نمی‌کند). پس از اصابت یک گلوله از این برج به بادکنک، آن بادکنک می‌میرد.

برج یخی



گلوله‌های این برج، یخی هستند! با اصابت هر گلوله یخی به یک بادکنک، آن بادکنک برای زمان مشخصی یخ می‌زند و ثابت می‌ماند (ولی نمی‌میرد) و سپس به حرکتش ادامه می‌دهد.

الگوریتم شلیک این برج، تقریباً همانند برج معمولی است؛ با این تفاوت که از بین بادکنک‌های موجود در دامنه شلیکش، به نزدیکترین بادکنک که یخ نزده است شلیک می‌کند (طبعاً اگر بادکنک یخ‌نzedهای در دامنه شلیکش وجود نداشت، شلیک نمی‌کند).

برج بمبی



گلولهای برج بمبی فقط یک بادکنک را نمی‌کشنند! بلکه تمامی بادکنک‌هایی که در شعاع مشخصی از محل فرود بمب هستند را نمی‌کشنند!

الگوریتم پرتاپ برج بمبی به این صورت است:

- بین تمامی بادکنک‌هایی که در دامنه شلیک این برج هستند، تعداد بادکنک‌هایی که از آن بادکنک، حداقل به اندازه دامنه آسیب بمب فاصله دارند را محاسبه می‌کند (اگر بادکنکی در دامنه شلیکش وجود نداشت، شلیک نمی‌کند).
- بادکنکی را انتخاب می‌کند که این مقدار برای آن بیشینه باشد.
- بمب را به بادکنک انتخاب شده شلیک می‌کند.

در واقع با این الگوریتم، سعی می‌شود بیشترین آسیب ممکن از پرتاپ یک بمب به دست بیاید.

بادکنک‌ها

بادکنک‌ها نیز همانند برج‌ها، انواع مختلفی دارند که تفاوت آن‌ها، در نحوه مردن آن‌هاست! دقت کنید که تمامی بادکنک‌ها در صورت دریافت گلوله یخی، نمی‌میرند و صرفاً به طور موقت یخ می‌زنند. توضیحاتی که در ادامه داده می‌شوند، مربوط به زمانیست که در دامنه آسیب بمب یا مورد اصابت گلوله برج معمولی قرار می‌گیرند.



بادکنک معمولی

بادکنک معمولی، ضعیفترین نوع دشمن شمامست! این نوع بادکنک‌ها با اولین گلوله از برج معمولی یا برج بمبی، می‌میرند و از صفحه بازی محو می‌شوند.

بادکنک باردار



این نوع بادکنک‌ها، دو بادکنک معمولی را باردار هستند. یک بادکنک باردار پس از مرگ محو می‌شود و فرزندان خود را از همان نقطه‌ای که قتل صورت گرفته است، به جنگ می‌فرستد.

پارامترها

در بازی، تعداد زیادی پارامتر وجود دارد:

- تمامی پارامترهای انواع مختلف بادکنک‌ها و برج‌ها
- میزان جان اولیه بازیکن
- میزان امتیاز اولیه بازیکن

وظیفه شماست که این پارامترها را جوری تنظیم کنید که محصول نهایی قابل بازی کردن باشد. همچنین، به منطقی بودن مقادیر خود توجه کنید. به طور مثال، کشنن بادکنک باردار، امتیاز بیشتری نسبت به کشنن بادکنک معمولی دارد (یا هزینه ساخت برج معمولی کمتر از برج یخی و بمبی است). پارامترهای شما باید در کد به نحوی تعریف شده باشند که به راحتی قابل تغییر باشند.

تنظیمات موج‌ها

علاوه بر سایر پارامترها، پارامترهای موج‌ها نیز باید توسط شما تنظیم شوند. با این تفاوت که برای تنظیمات موج‌ها، باید دقیقاً از ساختار زیر استفاده کنید (باید این ساختار را در فایلی در کدتان داشته باشید):

```
#ifndef WAVE_CONFIGS_HPP_INCLUDE
#define WAVE_CONFIGS_HPP_INCLUDE

#include <vector>
#include <map>
#include <string>

struct AttackWave {
    // Number of attacker balloons per type.
    std::map<std::string, int> enemiesCount;
    // When set to [100, 200], the gap time between the launch of every two
```

```

    // consecutive balloons should be random between 100ms and 200ms.
    std::pair<int, int> enemyLaunchGapMs;
};

const std::vector<AttackWave> ATTACKING_PLAN = {
    {{{"Normal", 5}}, (500, 700)},
    {{{"Normal", 10}, {"Pregnant", 5}}, (400, 600)},
    {{{"Normal", 15}, {"Pregnant", 10}}, (200, 400)},
};

// When set to 10, the first wave is launched at t = 10s, and the next one
// starts 10 seconds after the end of the previous wave.
const int WAVE_LAUNCH_GAP_SECS = 10;

#endif // WAVE_CONFIGS_HPP_INCLUDE

```

موسیقی

همچنین پس از شروع بازی آهنگی (ترجیحاً حماسی!) در پس زمینه پخش می‌شود تا حس غرور و رقابت شما را برانگیزد!

نکات تكميلی

- در اين تمرین تعدادي پaramتر وجود دارد (مانند سرعت حرکت بادکنكها، سرعت شليک تيرها، ميزان صدهه و قدرت انواع مختلف بادکنكها و ...). از شما انتظار می رود که طبق صلاحدي خودتان اين اعداد را تنظيم کنيد، به صورتی که بازي قابل بازي کردن باشد و تفاوت بخشهاي خواسته شده مشخص باشد (خيلي روی اعداد دقیق حساس نباشید). در صورت تمرین کليت انتظارات از بازي بيان شده و در تصميمگيري شيوه پياده‌سازی آنها (مانند فيزيك بازي، اندازه اشيا و ...) دست شما باز است. هدف نهايی اين است که برنامه نهايی قابل بازي کردن باشد.
- برای اينکه در زمان تحويل برای دستیار مربوطه مشکلی ایجاد نشود ابعاد صفحه را حداکثر $720 * 1080$ در نظر بگيريد.
- توصيه می‌شود که پروژه را از ساده به پيچيده جلو ببريد و ابتدا کليت بازي را پياده‌سازی کنيد. به عنوان مثال می‌توانيد ابتدا فقط حرکت بادکنكها در صفحه را بنويسيد. سپس در طی چند مرحله، برنامه را پيچيده‌تر کنيد. همچنان روند ارزیابی شما به صورتی قسمت‌بندی شده است که حتی در صورتی که پروژه را كامل نکرده باشيد، نمره مربوط به قسمت‌های كامل شده به شما تعلق بگيرد.
- دقت کنيد که تصميمگيري در خصوص مواردي که به طور صريح در صورت تمرين ذكر نشده، بر عهد شما بوده و می‌توانيد هر فرضی که می‌خواهيد را پياده‌سازی کنيد.
- از آنجايي که پروژه باید به صورت چند فايلی پياده‌سازی شود، توصيه می‌شود که پروژه را از ابتدا در چند فايل پياده‌سازی کنيد و از پياده‌سازی کل پروژه در يك فايل و سپس تبديل آن به چند فايل خودداری کنيد.
- توجه داشته باشيد که در اين تمرین سبك برنامه‌نويسی رويداد محور را در ساختار کد خود در نظر بگيريد.

نحوه تحويل

- میتوانید این تمرین را در قالب گروههای دو نفره انجام دهید. گروهی انجام دادن این تمرین، الزامی نیست و میتوانید به صورت تکی نیز آن را تحويل دهید. در صورتی که تمرین را به صورت گروهی انجام میدهید اسامی اعضای گروه را در این [شیت](#) وارد کنید.
- توجه داشته باشید که در صورت گروهی انجام دادن تمرین، هر دو عضو گروه باید اشراف کامل روی تمرین و کد آپلود شده داشته باشند و این موضوع در تحويل تمرین در نظر گرفته میشود. در صورت عدم رعایت این مورد، نمرهای برای فردی که روی کد اشراف ندارد، در نظر گرفته نمیشود. همچنین، هر دو نفر باید مشارکت کافی در انجام پروژه را داشته باشند؛ این موضوع از طریق commit ها در گیت، بررسی خواهد شد.
- در صورت انجام تمرین به صورت گروهی، آپلود تمرین از طرف یکی از اعضای گروه کفایت میکند.
- تحويل این تمرین در سامانه گیت‌هاب انجام میشود. برای انجام تمرین لطفاً از طریق [این لینک](#) وارد شوید، پس از آن اگر یک گروه ساخته و در صورت گروهی انجام دادن تمرین، همگروهی شما نیز باید از طریق لینک عضو همان گروه شود (دقیقت کنید که اگر انفرادی تمرین را انجام میدهید نیز بایستی یک تیم بسازید، همچنین توجه کنید که اگر همگروهی شما از قبل گروه را ساخته عضو گروه دیگری نشوید). پس از آن به صفحه‌ای منتقل میشوید که در آنجا میتوانید تمرین جدید را قبول کنید، پس از قبول کردن تمرین یک مخزن⁴ در [AP Assignments](#) برای شما ساخته میشود و باید کدهای خود را در آنجا قرار دهید.
- پس از انجام تمرین و بارگذاری در گیت‌هاب، کد Hash آخرین کامیت⁵ را به همراه شماره دانشجویی خود در سامانه ایلرن آپلود کنید (در خط اول شماره دانشجویی، پس از آن از Enter استفاده کنید و به خط بعد بروید و پس از آن Hash آخرین کامیت). نمونه متن خواسته شده در سامانه ایلرن (بخش `<sid> <last_commit_hash>`):

```
<sid>
<last_commit_hash>
```

: نمونه

```
810100000
bad8fbcdcf3b9feb371a31e0c370150aa870b18
```

⁴ Repository

⁵ Commit

- دقت کنید که عدم رعایت ساختار گفته شده در آپلود یا تغییر ساختار فایل‌ها در مخزن (می‌توانید به دلخواه خود فایل اضافه کنید و ... اما اسم و ساختار فایل‌هایی که در ابتدا به شما داده می‌شود نباید تغییر کند) باعث کسر 5 درصد از نمره شما خواهد شد.
- پروژه شما باید به صورت چند فایل⁶ و با استفاده از makefile پیاده‌سازی شده باشد. هدف اصلی پروژه پیاده‌سازی شی‌گرایی بوده و پیاده‌سازی به صورت چند فایلی صرفا برای آشنایی شما با این مفهوم می‌باشد. دقت کنید در پروژه‌های بزرگ‌تر، شما از ابتدا باید فایل‌ها و اجزای مختلف پروژه را تشخیص داده و آن را پیاده‌سازی کنید؛ با این حال برای سادگی بیشتر و به منظور کسب تجربه، بهتر است این پروژه را ابتدا به صورت یک فایل پیاده‌سازی کرده و تمرکز خود را روی طراحی شی‌گرا بگذارید؛ پس از تشخیص کلاس‌ها و پیاده‌سازی پروژه، آن را به چند فایل تقسیم کرده و makefile مناسب را بنویسید.
- دقت کنید که فایل makefile باید در صفحه اول مخزن باشد و در پوشه‌ای قرار نداشته باشد و در آن مشخص کنید که از استاندارد C++ 20 استفاده می‌کنید.
- سعی کنید مرحله به مرحله کد را commit کنید و از commit کردن کل کد در انتهای پروژه خودداری کنید. همچنین، پیام commit-های شما باید خوانا باشد و تغییر ایجاد شده را توضیح دهد. دقت کنید که بخشی از نمره شما را استفاده صحیح از گیت تشكیل می‌دهد. برای مطالعه در خصوص نحوه صحیح commit کردن، می‌توانید [این لینک](#) را مطالعه کنید.
- نام برنامه قابل اجرای شما باید TD (بدون هیچ پسوندی مانند exe یا out) باشد و پس از ساخته شدن در کنار makefile قرار بگیرد (داخل پوشه‌ای فایل خروجی ساخته شده را قرار ندهید).
- حتماً فایل‌هایی که در پروژه استفاده شدند (مانند عکس، موسیقی و ...) را در مخزن قرار دهید، پروژه شما باید در سیستم دستیاران قابل اجرا شدن باشد.
- برای ایجاد رابط کاربری گرافیکی⁷ و تمامی افکتهاي برنامه خود باید از کتابخانه SFML استفاده کنید.
- مخزن شما باید پوشه‌ی کامل پروژه باشد که شامل کد کامل برنامه شما به همراه تصاویر و سایر موارد است.
- در این تمرین بازی شما توسط دستیاران آموزشی آزموده می‌شود و آزمون خودکار ندارد.
- نمره هر بخش در صورت صحت عملکرد آن در بازی شما اختصاص می‌یابد و داشتن کد یک بخش که در بازی قابل آزمودن نیست نمره‌ای برای شما ندارد.
- برنامه شما باید حتماً طراحی شی‌گرا باشد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

⁶ multi-file

⁷ GUI

نمرات

- تمیزی کد
 - رعایت کردن نامگذاری صحیح و انسجام⁸
 - عدم وجود کد تکراری
 - رعایت دندانه‌گذاری⁹
 - عدم استفاده از متغیرهای گلوبال
 - استفاده صحیح از متغیرهای ثابت به جای Magic Value-ها
- بازی
 - پیاده‌سازی موارد ذکر شده در صورت پروژه
- طراحی
 - کاهش Coupling
 - رعایت Encapsulation
 - طراحی مناسب چند فایلی
 - استفاده از Header Guard-ها
 - استفاده صحیح از Makefile
 - ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند
- گیت
 - پیام‌های خوانا برای commit-ها
 - کردن مرحله به مرحله Commit
 - مشارکت همه اعضای گروه و اشراف کامل بر روی کد

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

⁸ Consistency

⁹ Indentation